

Amendments To Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A method comprising:
transferring, via radio frequency (RF) communication, software directly to a software-defined radio device from a software server to create transferred software, said software server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software currently running in said software-defined radio device, and wherein said transferred software is stored in to at least a portion of a data store associated with said software- defined radio device; and
sending an instruction via RF communication directly to said software-defined radio device identifying a ~~selected software application which specifies whether~~ said transferred software or said software currently running in said software-defined radio device as a selected software application to ~~will~~ be loaded by said software-defined radio device during in response to a restart of said software-defined radio device, device; and
automatically switching from said selected software application to a different version of said selected software application, wherein the different version is selected based on a detected error
~~wherein said selected software application is loaded to said software-defined radio device during a restart of said software-defined radio device.~~
2. (Cancelled)
3. (Previously Presented) The method according to claim 1, further comprising monitoring said transferring of said transferred software and monitoring said loading of said selected software application.
- 4 – 5. (Cancelled)

6. (Previously Presented) The method according to claim 1, wherein said instruction identifies a software version.
7. (Previously Presented) The method according to claim 1, wherein said software-defined radio device comprises a plurality of software defined radio devices.
8. (Previously Presented) The method according to claim 1, further comprising receiving an error indication in response to a fault being detected in at least one of said transferring of said transferred software or said loading of said selected software application.
9. (Original) The method according to claim 1, wherein said transferred software comprises a plurality of software components.
10. (Previously Presented) The method according to claim 1, further comprising receiving a version indicator from said software- defined radio device, said version indicator identifying software which is currently loaded on said software-defined radio device.
11. (Previously Presented) The method according to claim 1, further comprising receiving a software listing from said software-defined radio device, said software listing identifying software currently available on said data store.
12. (Currently Amended) The method according to claim 1, wherein said transferred software is stored in [[to]] a second data store associated with said software-defined device.
13. (Previously Presented) The method according to claim 12, wherein said second data store is nonvolatile.
14. (Previously Presented) The method according to claim 1, wherein said transferring of said transferred software occurs in response to said software-defined radio device continuing to perform software-defined radio functions.

15. (Currently Amended) The method according to claim 1, wherein said software server comprises ~~[[is]]~~ a computer operatively connected to said software-defined radio device via a wireless communications network.

16. (Currently Amended) A method comprising:

receiving, via radio frequency (RF) communication directly from a software server, transferred software at a software-defined radio device, said software server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software currently running in said software-defined radio device, and wherein said software currently running in said software-defined radio device is stored in a first non-volatile data store area;

storing said transferred software in ~~to at least a portion of a~~ second non-volatile data store area distinct from said first non-volatile data store area ~~associated with said software-defined radio device;~~

receiving, via radio frequency (RF) communication directly from said software server, an instruction at said software-defined radio device identifying ~~a selected software application specifying whether~~ said transferred software or said software currently running in said software-defined radio device ~~currently running software~~ as a selected software application to ~~will be~~ loaded by said software-defined radio device during in response to a restart of said software-defined radio device;

responsive to a restart instruction, restarting said software-defined radio device and loading said selected software application; and

verifying said selected software application is loaded successfully.

17. (Currently Amended) The method according to claim 16, further comprising automatically reverting from said selected software application to a previous different software version without an instruction from said software server, wherein said automatically reverting is in response to ~~a fault being detected in said loading step~~ said selected software application encountering an error which causes said software-defined radio device to stop functioning properly.

18. (Currently Amended) The method according to claim 17, further comprising providing an error indication in response to said fault detection, and selecting said different software version based on said error.

19. (Previously Presented) The method according to claim 16, further comprising:
monitoring said receiving transferred software step; and
providing an error indication in response to a fault being detected in said receiving transferred software step.

20. (Previously Presented) The method according to claim 16, further comprising providing a version indicator to a remote location, said version indicator identifying software which is currently loaded on said software-defined radio device.

21. (Previously Presented) The method according to claim 16, wherein said selected software application identifies a software version.

22. (Currently Amended) The method according to claim 16, further comprising providing a software listing to a remote location, said software listing identifying software currently available on ~~said data store~~ said first non-volatile data store area.

23 - 24. (Cancelled)

25. (Previously Presented) The method according to claim 16, further comprising in response to receipt of said transferred software, decompressing said transferred software.

26. (Previously Presented) The method according to claim 16, wherein receiving said transferred software occurs while said software-defined radio device continues to perform software-defined radio functions.

27. (Currently Amended) A software-defined radio device comprising:

an RF communications interface configured to receive transferred software and an instruction directly from a software server remotely located with respect to the [[a]] software-defined radio device, wherein said transferred software is another version of software configured to be currently running in said software-defined radio device, and wherein said software server comprises a man-machine interface configured to receive ~~for receiving~~ from a system operator said instruction comprising a selected software configured to specify ~~specifying~~ whether said transferred software or said software configured to be currently running in said software-defined radio device ~~currently running software~~ will be loaded at in response to a restart of said software-defined radio device;

a data store associated with said software-defined radio device configured to store said transferred software in ~~on~~ at least a portion of said data store; and

a processor programmed to:

load said selected software to said software-defined radio device in response to ~~during~~ said restart of said software defined radio device; and

automatically revert, without an instruction from said software server, from said selected software to a different ~~previous~~ software version responsive to at least one of a fault in said loading of said selected software encountering an error which causes said software-defined radio device to stop functioning properly, or exceeding a predetermined number of attempts to successfully load said selected software.

28. (Previously Presented) The device according to claim 27, wherein said processor is further programmed to determine that said software and said instruction are received successfully and to determine that said selected software is loaded successfully.

29. (Cancelled)

30. (Previously Presented) The device according to claim 27, wherein said processor is further programmed to decompress said transferred software, and wherein said software server further comprises a compression application for compressing said software prior to said software being transferred.

31. (Previously Presented) The device according to claim 27, wherein said transferred software comprises a plurality of software components.

32. (Currently Amended) The device according to claim 27, wherein said RF communications interface is further configured to transmit a version indicator identifying said software configured to be currently running in said software-defined radio device ~~software which is currently loaded on said software-defined radio device~~ to said software server.

33. (Previously Presented) The device according to claim 27, wherein said RF communications interface is further configured to transmit a software listing identifying software currently available on said data store to said software server.

34. (Previously Presented) The device according to claim 27, further comprising a second data store associated with said software-defined device configured to store said transferred software.

35. (Previously Presented) The device according to claim 34, wherein said second data store is nonvolatile.

36. (Currently Amended) The device according to claim 27, wherein said processor is further programmed to receive said software ~~is received~~ from said software server while said software-defined radio device performs ~~continues to perform~~ software-defined radio functions.

37-38. (Cancelled)

39. (Currently Amended) A tangible computer-readable medium having stored thereon computer-executable a plurality of instructions that, if executed by a computing device, cause the computing device to perform a method ~~said plurality of instructions~~ comprising:

~~instructions to receive~~ receiving, via radio frequency (RF) communication direct from a software server, transferred software at a software-defined radio ~~device~~ device, said software

server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software currently running in said software-defined radio device, and wherein said software currently running in said software-defined radio device is stored in a first data store area;

~~instructions to store~~ storing said transferred software ~~in to at least a portion of a second data store area distinct from said first data store area associated with said software-defined radio device;~~

~~instructions to receive~~ receiving via radio frequency (RF) communication direct from a software server an identification of ~~a selected software application specifying whether said transferred software or said currently running software~~ as a selected software application to will be loaded by said software-defined radio device in response to ~~during~~ a restart of said software-defined radio device;

~~instructions to restart~~ restarting said software-defined radio device and loading ~~load~~ said selected software, responsive to a restart instruction, while maintaining said software currently running in said software-defined radio device in said first data store area and said transferred software in said second data store area; and

~~instructions to verify~~ verifying said selected software application is loaded successfully.